

Evaluating OWL Options

For CMCS498w

April 26, 2004

Evaluating OWL

- Is OWL (or any of its subspecies) what we want from a Web Ontology Language?
 - Do we still *want* a Web Ontology Language?
 - Will OWL *scale* and *spread*?
- How do we know when to use (which) OWL?
 - For vertical applications
 - For publishing ontologies on the web
 - For building common vocabularies
 - For saying certain things
- When to look for alternatives?

OWL Lite

- OWL Lite as a Description Logic
 - Almost SHIF(D) (technically, it's a variant of SHIN(D) with cardinalities restrictions of only 0 or 1)
 - Van Harmalan [argues](#) hits 80/20
 - For *conceptual* complexity
 - OWL lite is also computationally less complex (we think!)
 - Is it technically complex?
 - Syntax?
- OWL Lite is not a straightforward extension of RDFS
 - Different sort of conceptual complexity
- Allows for feature chains and part of relations

OWL DL

- As Description Logic
 - Mostly SHOIN(D)
 - Close to the boundaries of well understood DL expressiveness (SHIQ and SHOQ are well understood, (D) is fairly well understood)
 - Oddly missing Q
- Trades expressivity for decidability and (some) efficiency
 - Is this the [right tradeoff?](#)
- Some (hurting) missing bits:
 - Role chains, Horn rules, Non-monotonic constructs, Metaclasses, metamodeling, data validation, etc.

RDFS and OWL Full

- OWL Full *liberalizes* OWL-DL
 - Remove all sorts of syntactic and semantic restrictions
 - Classes can be properties can be individuals can be datatypes can be datavalues all at once
 - Number restrictions on transitive roles, etc.
 - This alone makes it semi-decidable
- Everything is a resource
 - And so can have attributes
 - This is very natural
 - The AnnotationProperty hack
- OWL Full supports a kind of [metamodeling](#)

OWL Full: Metamodeling, how?

- In FOL (and OWL-DL):
 - *Classes* are 1 place predicates (or formulas with one free variable):
 - $\text{Person} == \text{Person}(x)$
 - $(\text{conj}, \text{Person}, (\text{some}, \text{hasChild}, \text{Woman})) == \text{Person}(x) \ \& \ \exists(y)(\text{hasChild}(x, y) \ \& \ \text{Woman}(y))$
 - *Properties* are 2 place predicates
- What makes FOL *first-order* is that you can't *quantify* over predicates
 - No “For all 1 place predicates such that...”
 - No “There is a predicate, P , such that...”

Metamodeling, how? II

- But in RDF(S), we have only triples
 - Two place predicates?
 - How to understand `rdf:type`?
 - For subjects that are individuals
 - For subjects that are classes or properties
 - `C rdf:type owl:Class`
 - `P rdf:type owl:ObjectProperty`
 - How to understand `rdfs:subClassOf`?
- In OWL-DL, triples are *surface syntax*
 - $(a, \text{rdf:type}, B) == B(a)$
 - $(C, \text{rdfs:subClassOf } D) == C \rightarrow D$
- In OWL Full, the surface syntax is part of the domain

Metamodeling, how? III

- In OWL Full, classes are individual, i.e., *domain elements*
- They have relations to other individuals
 - E.g., `rdf:type`
 - That relation is true just in case the potential member is a member of the class's *extension*
 - The extension is just another object related to the class
- Thus, classes can be self-members without fear of paradox (in the simple case)
- We recover the normal semantics of membership with axioms:
 - $(a, \text{rdf:type}, C) \text{ iff } C'(a)$

Metamodeling, how? IV

- Prior technique requires *deleting* the triple
 - Thus, one hopes, capturing, say, OWL-DL exactly
- This eliminates the metamodeling
- Another approach: second order syntax, first order semantics:
 - Let every triple be represented by a three-place predicate (triple/Property Value/rdf)
 - Add axioms which define analogous entailments
 - (C, subClassOf, D) & (D, subClassOf, E) then
(C subClassOf E) (rdfs11)
 - (a, type, C) & (C, subClassOf, D) then
(a, type, D) (rdfs9)

OWL ontologies as interlingua

- What *is* an *Ontology* (or Web Ontology)?
 - What axioms constrain: `:foo rdf:type owl:Ontology`?
 - What should non-reasoners do with Web ontologies?
- What is a *Class*?
 - An object with an extension?
 - An extension/set?
 - Something with an associated definition?
 - Which is prior, the class or its definition(s)?

Connecting Ontologies

- How do we get to a *Web* of Ontologies?
 - What is it to *link* to an ontology?
 - What is it to *use* an ontology?
 - What does owl:imports give us?
- How do we *map* between Ontologies?
 - Are the OWL constructs enough?
- How do we express agreement or disagreement?

Case Study: Web Services

- Alternative vision of the program friendly Web
- Focus on Web *dynamics*, i.e., behavior
- Requires rich but “machine processible” descriptions
- Enter (Semantic (Web) Services)
 - Services for the Semantic Web
 - Semantic Web enabled Web Services
- Rich domain with lots of automatable tasks
 - Discovery, composition, execution, monitoring
- Lot of extent activity
 - OWL-S, SWSI, WSMO, many research efforts

Reading

- OWL-S: Semantic Markup for Web Services
 - <http://www.daml.org/services/owl-s/1.1/owl-s.html>
- OWL-S Ontologies
 - <http://www.daml.org/services/owl-s/1.1/>
- WS Architecture Ontologies
 - <http://www.w3.org/2004/02/wsa/>
- What's in an instance?
 - <http://citeseer.ist.psu.edu/384751.html>
- RDFS(FA) and RDF MT: 2 Semantics for RDFS
 - <http://www.cs.man.ac.uk/~panz/Zhilin/download/Paper/Pan-Horrocks-rdfsfa-2002.pdf>
- What is a Knowledge Representation?
 - <http://medg.lcs.mit.edu/ftp/psz/k-rep.html>